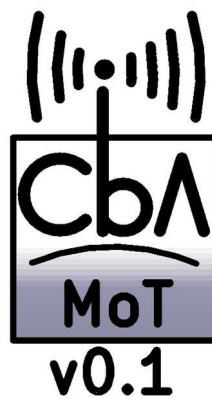




Carrera: Ingeniería Electrónica

Proyecto:

CbA-MoT



Módulo inalámbrico para monitoreo de sensores remotos

## Especificación Funcional

**ID 1883**

Autor  
Sebastián Allende  
[cballende/TDSTec.git](https://github.com/cballende/TDSTec)

Fecha: 20/ 8/ 2022

Versión 1.0



## Especificación funcional

---

Fecha	Versión	Descripción	Autor/a
20/ 8 / 2022	1.0	Versión inicial	Sebastián Allende <a href="https://github.com/cballende/TDSTec.git">cballende/TDSTec.git</a>

## Índice

1. Introducción.....	5
1.1. Propósito del documento.....	5
1.2. Alcance del proyecto.....	5
1.3. Personal involucrado.....	5
1.4. Definiciones, acrónimos y abreviaturas.....	6
1.5. Referencias.....	6
1.6. Resumen.....	6
2. Descripción del dispositivo.....	6
2.1. Bloques de hardware del dispositivo.....	7
2.2. Bloques de software del dispositivo.....	9
3. Especificaciones funcionales.....	12
3.1. RF01: Ahorro de energía.....	12
3.2. RF02: Asociar a red.....	13
3.3. RF03: Conexión indirecta.....	14
3.4. RF04: Funcionar como repetidor.....	14
3.5. RF05: Asociar y autenticar con administrador de red.....	15
3.6. RF06: Almacenamiento de datos.....	16
3.7. RF07: Actualización remota.....	17
3.8. RF08: Almacenar variables normales.....	18
3.9. RF09: Enviar datos automáticamente a servidor.....	19
3.10. RF10: Interacción y reacción con administrador.....	20
4. Requisitos no funcionales.....	27
Cualidades ejecución.....	28
Cualidades evolución.....	28
Otros requisitos.....	28

### 1. Introducción

Este documento corresponde a la Especificación Funcional para la solución **CbA-MoT**. Esta especificación se ha estructurado basándose en la información mencionada en el documento de Especificación de Requisitos (ER) ID 1882.

#### 1.1. Propósito del documento

El presente documento tiene como propósito proveer información detallada de cómo funcionará el sistema, cuáles serán sus comportamientos deseados y cómo se deberá construir, con base en los requerimientos anteriormente definidos en la E.R.

Está dirigido a:

- Los desarrolladores del software.
- Los desarrolladores del hardware.
- Los directores de trabajo final, quienes evaluarán su funcionamiento.

#### 1.2. Alcance del proyecto

El proyecto involucra el diseño de un dispositivo, un módulo stick que debe ser capaz de procesar señales de sensores asociados, filtrar comportamientos no deseados y detectar eventos, posteriormente enviar datos en un paquete normalizado hacia un recolector o dispositivo maestro remoto.

El instrumento desarrollado en este trabajo final se trata de un prototipo para pruebas en campo de una actividad productiva de un cliente.

El algoritmo de detección de eventos difiere según la aplicación de interés del cliente, lo que obliga a normalizar la interfaz entre el procesamiento de datos del sensor específico y la aplicación principal del software en la microcomputadora.

El dispositivo debe ser capaz de asociarse a un red, enviar datos, recibir comandos, auspiciar de mediador entre otros dispositivos iguales, poseer modo ahorro energía para ser viable la alimentación mediante baterías o pilas.

El tamaño y la tecnología de fabricación del dispositivo son restringidas, prima utilizar encapsulados comerciales de la menor superficie para los componente identificados según los criterios de funcionales y disipación de energía. Seguido, el diseño de pcb debe ser multicapa y montaje sobre una sola faz.

El diseño debe lograrse con el objetivo de brindar la base para un salto o instancia hacia una versión de producto mínimo viable. Por definición se adicionaran funcionalidades de segunda categoría progresivamente durante su uso por parte del cliente.

#### 1.3. Personal involucrado

Nombre	Sebastián Allende
Rol	Diseñador
Categoría Profesional	Diseñador de sistemas embebidos
Responsabilidad	Diseño del sistema

## 1.4. Definiciones, acrónimos y abreviaturas

Nombre	Descripción
RF	Requisito Funcional
RNF	Requisito No Funcional
MoTe	Sensor remoto
Estación	Equipo que tiene capacidad de comunicaciones.
BSS	Conjunto de Servicio Básico
M.L.A	Biblioteca para aplicaciones Microchip
RFD/FFD	Dispositivo de reducida/completa funcionalidad.
IDLE	

## 1.5. Referencias

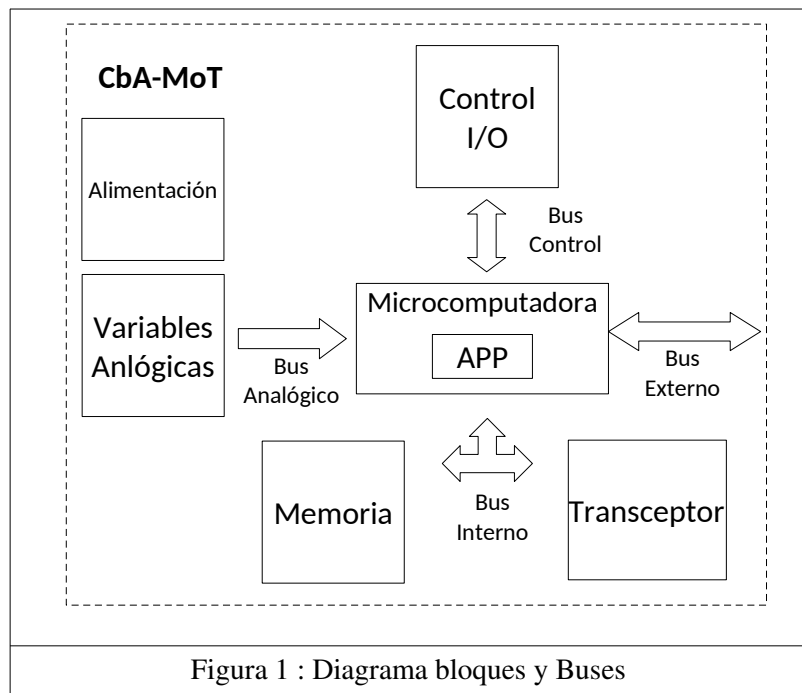
Título del Documento
<a href="#">Especificaciones de Requisitos</a>

## 1.6. Resumen

El presente documento tiene 4 secciones, la primera es introductoria, la segunda es de carácter global detallando los principales aspectos de la solución y las restantes secciones se profundiza con mayor detalle sobre las funcionalidades particulares.

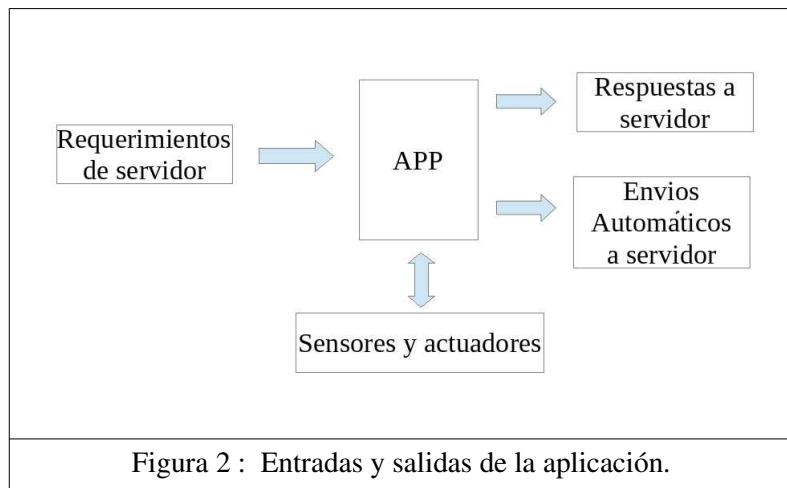
## 2. Descripción del dispositivo

El dispositivo a desarrollar es un módulo inalámbrico de monitoreo de sensores remotos, se compone de una microcomputadora de 8 bits de palabra, transceptor de radio frecuencia, bus de datos externo, bus de datos interno, bus de control interno, almacenamiento de datos volátil y no volátil, convertidor de tensión, regulador de tensión, sensado de variables analógicas internas.

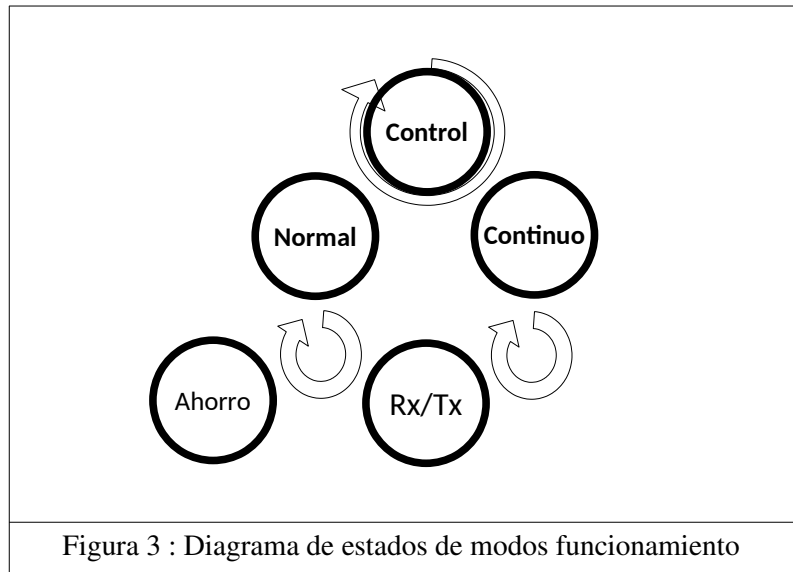


Las entradas y salidas de aplicación implementada en el firmware de la microcomputadora denominada como APP son definidas por dos elementos, primero el administrador que ejerce control sobre el dispositivo y segundo por la interacción con el medio en donde se encuentra ubicado el dispositivo que son mediante los sensores y actuadores.

La aplicación es un sistema reactivo que evoluciona según eventos hacia estados definidos, ellos permite usar bibliotecas MLA de MCHP a diferencia de implementar un RTOS como se describe en la nota AN1264 de MCHP modificando las MLA.



La lógica de control de APP agrupa el funcionamiento del dispositivo en dos modos, modo **Normal** donde el dispositivo funciona de forma automática con ahorro de energía y modo **Continuo** donde el dispositivo funciona a demanda, sin ahorro de energía y controlado por administrador en tiempo real.



## 2.1. Bloques de hardware del dispositivo.

### a) Microcomputadora

Se encarga de correr la aplicación APP, se vinculado a todos los elementos del dispositivo de la figura 1 mediante buses específicos de datos y de control .

### b) Transceptor

Se encarga de establecer un medio de comunicaciones inalámbrico entre dispositivos y el administrador.

### c) Memoria

Se encarga de almacenar datos de interés provenientes de los sensores, dispositivos asociados inalámbricos y del administrador.

Los datos corresponden a información que APP requiere para interactuar con el administrador una vez que se haya efectuado la asociación del dispositivo en la red.

La memoria esta dividida en dos bloques, uno no volátil y el otro volátil.

La memoria no volátil su utilidad es requerida para el sistema de red, datos relevantes del estado actual del dispositivo y de los sensores.

La memoria volátil es útil para el procesamiento grandes cantidades de datos de la etapa de adaptación de los sensores, tablas de rutéo extendidas y para almacenar el nuevo firmware que proviene del administrador que se utilizara por la funcionalidad de bootloader.

### d) Control IO

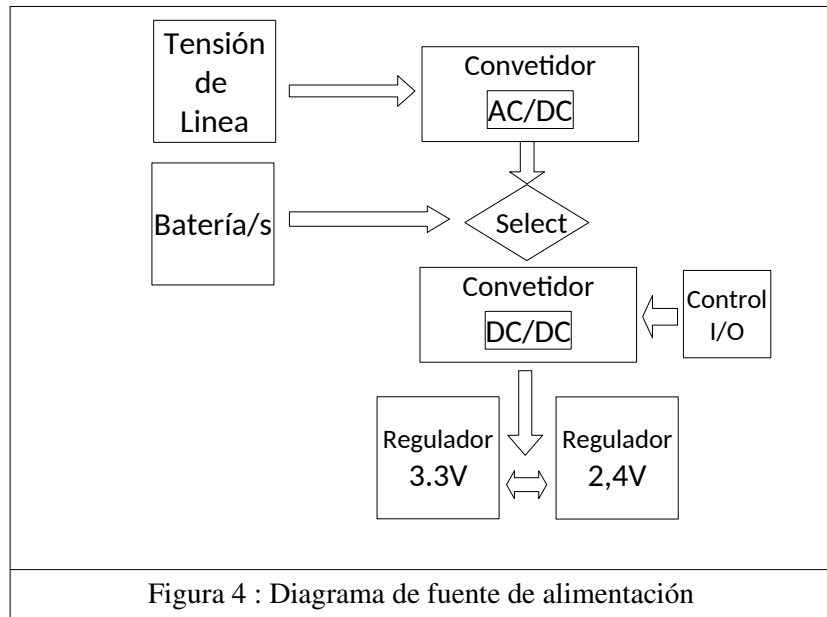
El bus de control tiene como fin administrar la habilitación los recursos periféricos como memorias, sensores, transceptor, indicadores testigos como leds, convertidor de tensión y reset externo del dispositivo.

### e) Variables analógicas

Las variables analógicas corresponden a indicadores internos del dispositivo como son la tensión de alimentación, la temperatura, el consumo de corriente del dispositivo.

### f) Etapa de alimentación.

La alimentación del dispositivo puede ser efectuada mediante pila o mediante tensión domiciliaria 220Vrms.

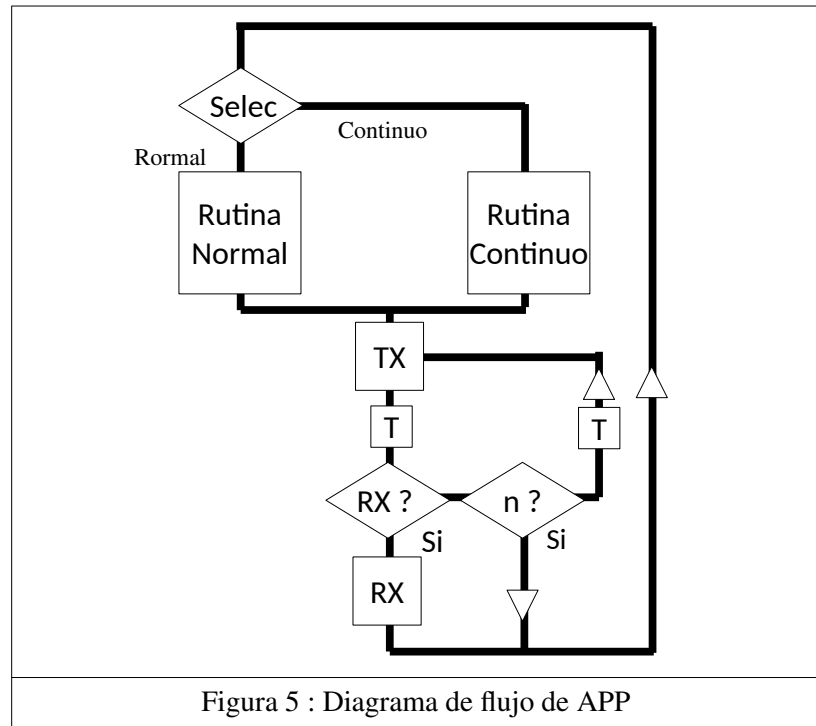


## 2.2. Bloques de software del dispositivo

### a) APP

La aplicación principal se encarga de administrar todos recursos. Se definen dos estados principales de funcionamiento **Normal** y **Continuo**. En modo Normal se emplea ahorro de energía y por lo tanto si las comunicaciones son efectuadas en periodos regulares ajustables. En el caso de modo Continuo las comunicaciones son bajo una probabilidad productos de eventos.





## b) Rutina Normal

La rutina de aplicación Normal tiene como objetivo producir un ahorro de energía para maximizar el uso de baterías, como consecuencia las comunicaciones se efectuaron en periodos regulares que son múltiplos del periodo de estado de ahorro de energía.

Los eventos de interés son reducidos y se discriminan como la funcionalidad principal del dispositivo, la ausencia de definición de eventos de interés vuelven al dispositivo sin utilidad.

Los eventos no disparan la salida de estado de ahorro de energía, los eventos son contabilizados por un sistema independiente del mecanismo de ahorro de energía y no requieren procesamiento. Los eventos son datos de interés que se enviaran al servidor.

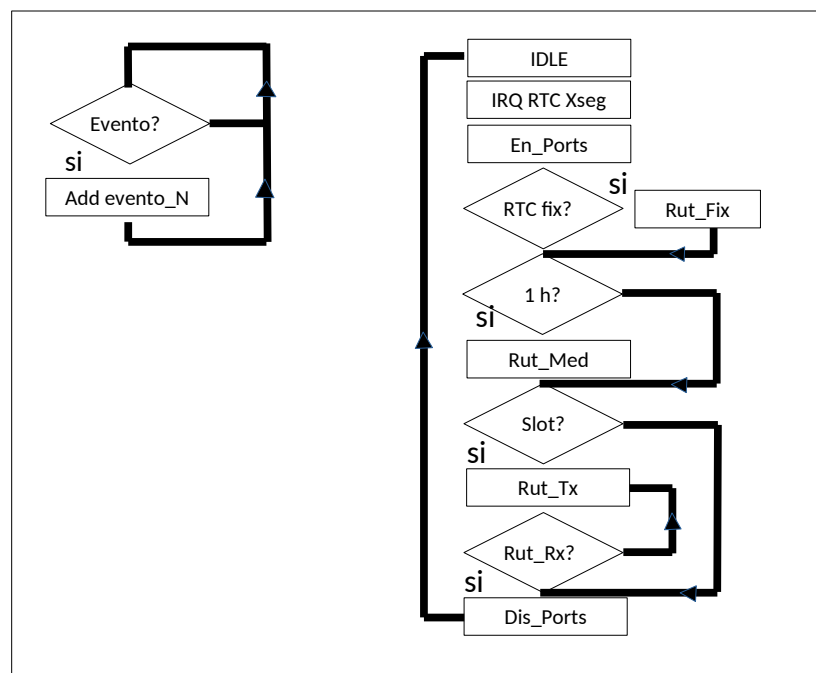


Figura 6 : Diagrama de flujo de rutina **Normal**

El sistema se sincroniza cada X cantidad de segundos de periodo de inactividad, cuando se vuelve activo el sistema ejecuta rutinas de verificación de relojes que asegure integridad.

Cada 1 hora se debe determinar la cantidad de eventos ocurridos.

Si se encuentra en periodo de transmitir datos al servidor, transmite con un algoritmo de recogimiento (ACK). Finalizada la transmisión el dispositivo vuelve a modo de ahorro de energía.

Si se encuentra en periodo de recepción de datos, escucha las tramas de las estaciones vecinas. Finalizado el periodo el dispositivo vuelve a modo de ahorro de energía.

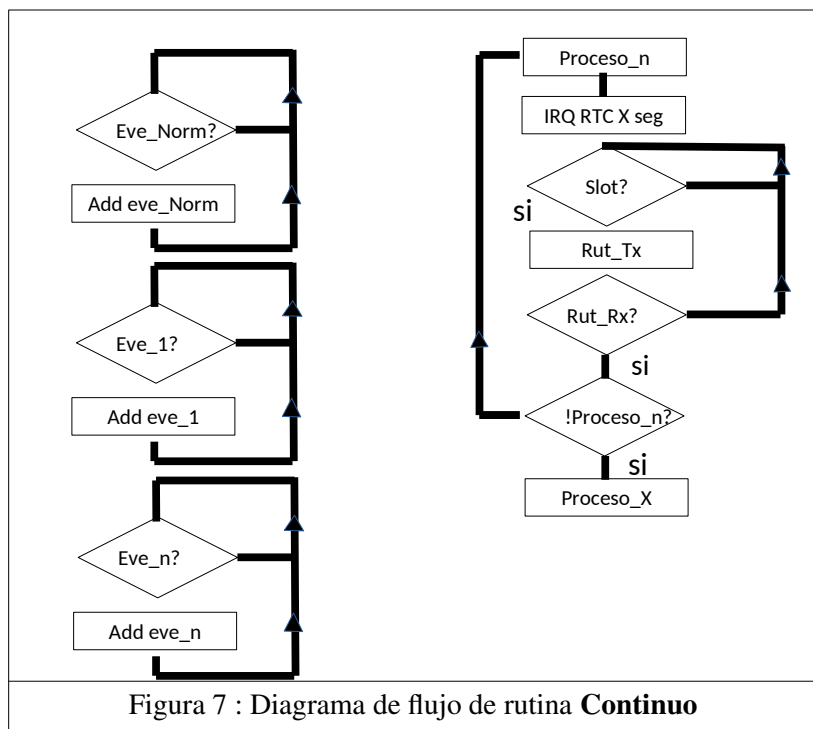
### c) Rutina Continuo

La rutina de aplicación **Continuo** tiene como objetivo producir una comunicación en tiempo real con el administrador , como consecuencia las comunicaciones deben iniciarse bajo contienda en el medio a transmitir.

Los eventos registrados y las actuaciones efectuadas en el dispositivo que tienen lugar en este estado no son la función principal del dispositivo. Los eventos normales también integran este modo a los que se incluyen los eventos propios que hacen al diagnostico del sistema y las actuaciones que forman parte de las funciones de administración y control.

Lo administrado en modo **Continuo** se encuentra normado, los hilos de ejecución implementados son únicos por funcionalidad. Esto quiere decir que cada proceso en modo continuo atiende un solo tipo de evento o actuación. El administrador remoto define que hilo ejecutar.

La comunicación por este modo establece una comunicación permanente hasta que el administrador la finalice.



### d) Rx/Tx

Las operaciones de recepción y transmisión contienen datagramas normalizados para la aplicación APP y encuentran encapsulados tramas 802.15.4 en su sección de payload.

Se establece que para cada tipo de operación de aplicación se considere un tipo de datagrama.

Tipos de datagramas:

**Normal**  
**Control**  
**Asociación**  
**Nop**

Las datagramas tienen los siguientes campos:

**Tipo de datagrama**  
**Modo:** Directo o no.  
**ID dispositivo**  
**Timestamp**  
**Ruido Rx**  
**Potencia Tx**  
**Nº de retransmisiones**  
**Clave :** permite acceso al Gateway.  
**Token:** permite acceso a la aplicación del servidor.  
**Secuencia :** número de datagrama.  
**Medidas :** vector de medidas.  
**Tiempo de medidas**  
**Inicio de transmisiones**  
**Datos brutos:** lista.

### 3. Especificaciones funcionales

EF 01	El dispositivo ahorra energía.
EF 02	El sistema se asocia a una red.
EF 03	El sistema establece una conexión con otros dispositivos remotos.
EF 04	El dispositivo brinda servicio de intermediario para las comunicaciones entre otros dispositivos.
EF 05	El dispositivo se asocia a un administrador.
EF 06	El dispositivo almacena resultados de las operaciones ejecutadas hasta obtener una conexión.
EF 07	El dispositivo se actualiza remotamente.
EF 08	El dispositivo envía datos y eventos.
EF 09	El dispositivo envía datos al un servidor de forma autónoma.
EF 10	El dispositivo responde y reacciona ante requerimientos del administrador

### 3.1.RF01: Ahorro de energía

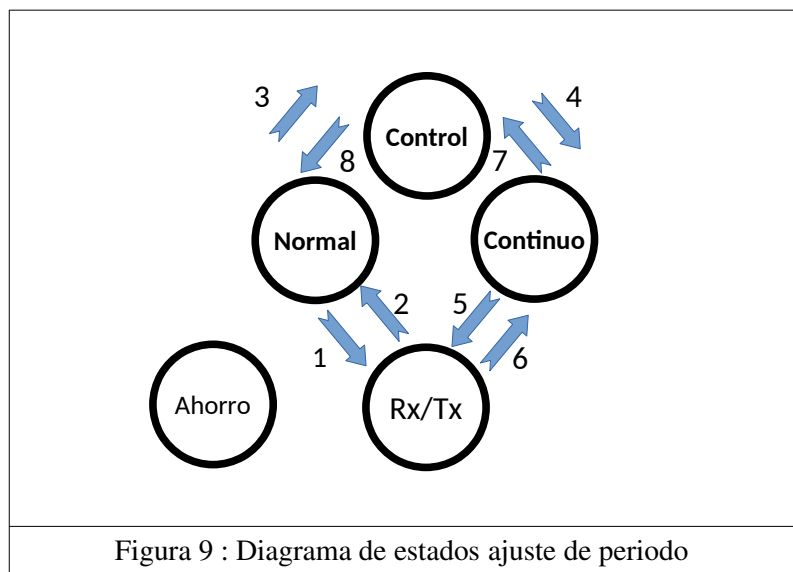
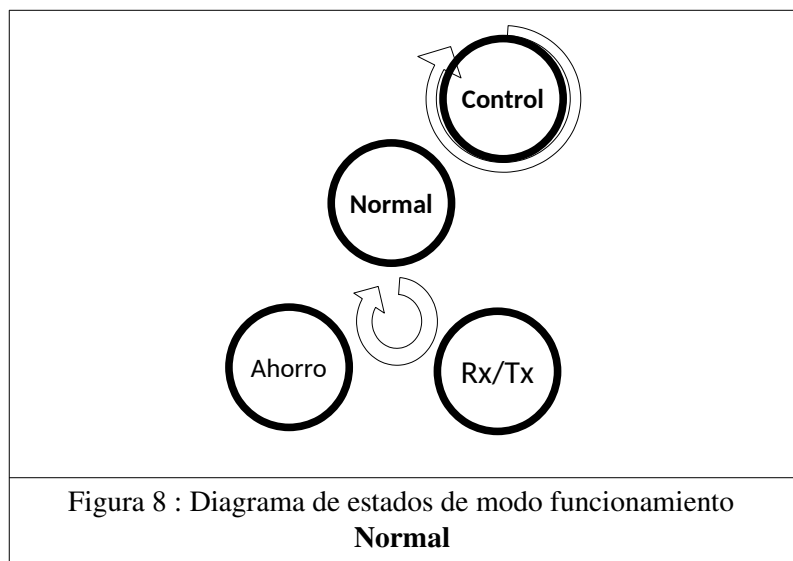
#### Descripción

El sistema ingresa y sale del modo de ahorro de energía periódicamente. El periodo que permanece en este estado esta establecido por defecto y es ajustable por el administrador.

El ajuste se encuentra asignado mediante una orden especial en modo **Continuo**.

El estado de ahorro de energía opera unicamente en modo **Normal**.

#### Esquema



#### Requisito Funcional

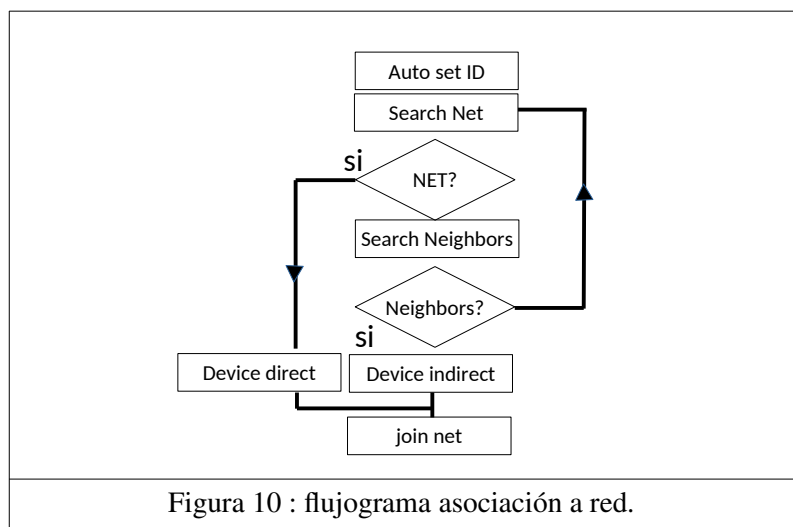
ID	Descripción	Referencia
RF01	Modo sleep	<a href="#">ER</a>

### 3.2. RF02: Asociar a red

#### Descripción

El dispositivo antes de ingresar a la aplicación principal busca redes activas para asociarse. De no haber redes disponibles busca asociarse a un elemento con su misma nivel jerarquía. Se asocia indirectamente a una red.

#### Esquema



#### Requisito Funcional

ID	Descripción	Referencia
RF02	Asociación a redes	<a href="#">ER</a>

### 3.3. RF03: Conexión indirecta

#### Descripción

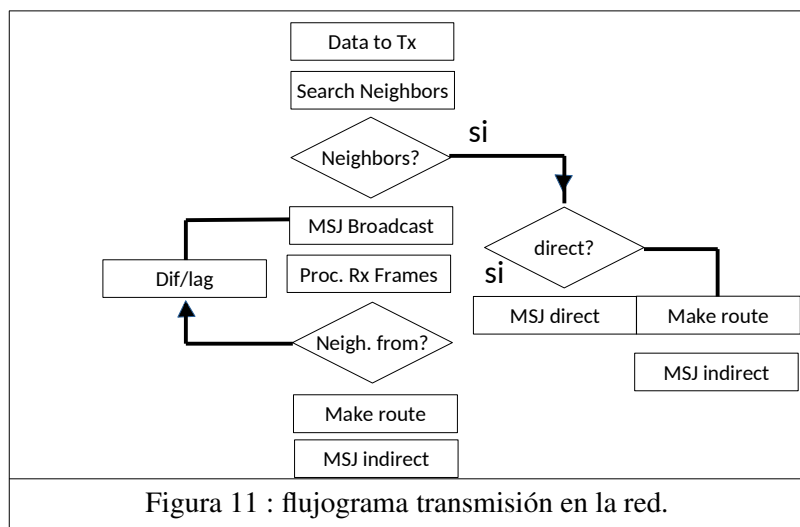
El dispositivo requiere iniciar una comunicación hacia un dispositivo fuera de alcance, se apoya en la existencia de dispositivos intermedios.

Primero busca en la tabla de dispositivos vecinos, en caso nulo busca en las tablas de vecinos sus dispositivos vecinos. En caso nulo nuevamente, emite un mensaje de broadcast que contiene información del nodo destino.

Los mensajes de respuesta de broadcast contienen la ruta hacia donde se proceso el paquete con respuesta afirmativa.

Se arma un paquete indicando mensaje indirecto, la dirección de destino y de próximo salto.

## Esquema



## Requisito Funcional

ID	Descripción	Referencia
RF03	Conexión indirecta	<a href="#">ER</a>

### 3.4. RF04: Funcionar como repetidor

#### Descripción

Un dispositivo requiere iniciar una comunicación hacia un dispositivo fuera de su alcance, se apoya en la existencia de dispositivos intermedios.

Cada dispositivo cuando se anuncia a sus vecinos incluye junto a su tabla de direcciones el momento permanecerá el, los integrantes de su tabla en etapa de escucha activa y la potencia a la cual transmiten.

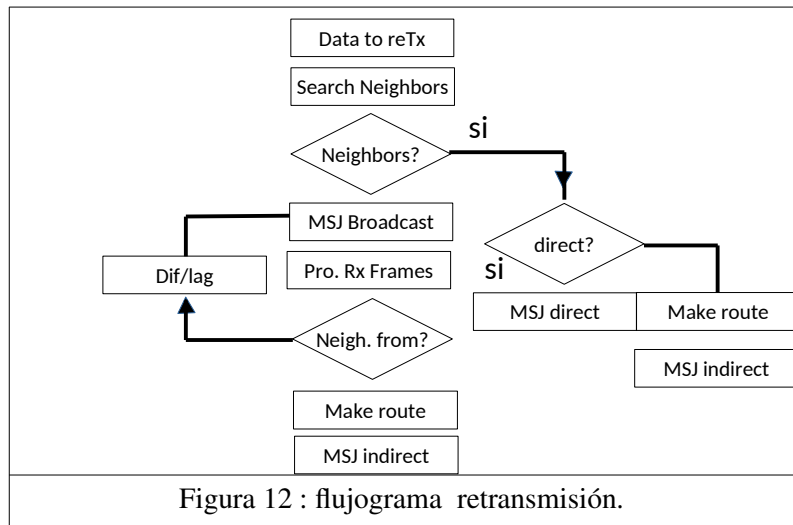
Esto permite coordinar cuando el mensaje sera transmitido, luego ir a modo sleep y despertar para recibirlo, también determina la ruta más corta al conocer la potencia con que transmiten.

Los mensajes recibidos son almacenados hasta que la estación de destino este en modo escucha, esto indica que el mensaje de broadcast no es por inundación y por única vez, el mensaje broadcast se emite de forma discrecional, en tiempos determinados por el intervalo de escucha de las estaciones destino y con tiempo de vida de 5 saltos (max 500 m).

Esto difiere del estándar Zigbee o 802.15.4 donde quien sincroniza la red es un punto de acceso o coordinador que son FFD. Aquí los RFD pueden retransmitir y coordinarse gracias a extensión de funcionalidad adicionada en APP. Lo que implica es aumento de latencia, mayor funcionalidad y ahorro de energía.

Es fácil imaginar que existirá una transmisión escalonada si las estaciones no están sincronizadas a despertar en el mismos intervalos de tiempo. Esto es útil para reducir la colisiones y la energía consumida. Se paga con reducción velocidad y throughput de la red, lo importante es extender el tiempo entre servicios de recambio de baterías.

## Esquema



## Requisito Funcional

ID	Descripción	Referencia
RF04	Repetidor	<a href="#">ER</a>

### 3.5. RF05: Asociar y autenticar con administrador de red

#### Descripción

El dispositivo para poder comunicarse con el administrador y el servidor requiere asociarse a la red y autenticarse con el administrador y el servidor.

Existen razones de seguridad que en esta descripción y la especificación técnica no se detallaran. Los mismos son descriptos en un documento privado.

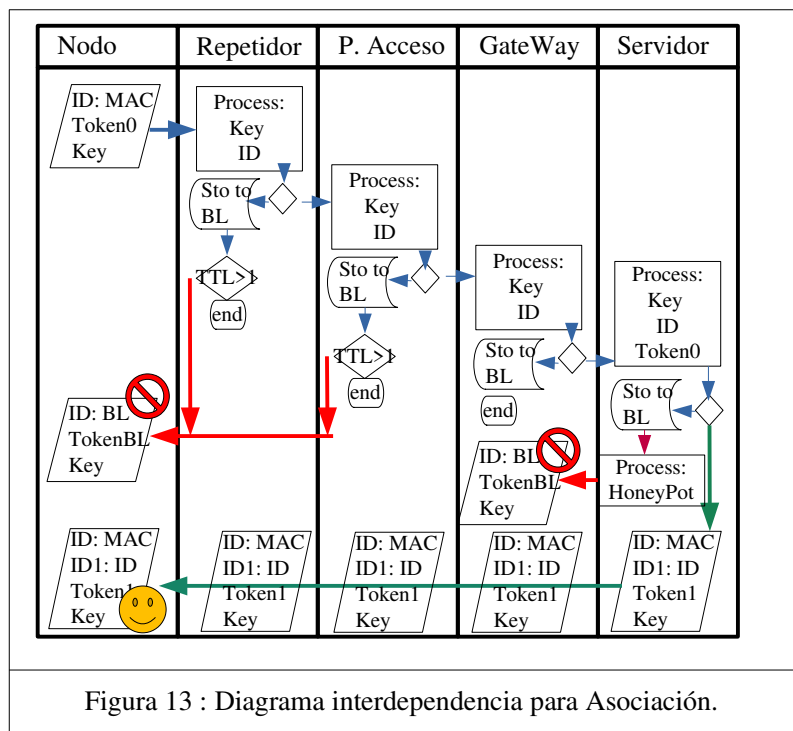
Hay que detallar que la asociación a la red la pila de protocolos lo hace de modo transparente con excepción de un dispositivo RFD que no se encuentra asociado a un coordinador sino que se conecta a otro RFD. Lo que se controla es el acceso al PA de la red, servidor y al administrador.

Se mitiga el ataque o la intromisión de nodos no permitidos dentro de la red mediante direcciones asignadas durante la asociación de ID blacklist y entorno de honeypot que son reconocidas por los dispositivos como tales y descartan los paquetes recibidos.

También así como segundo nivel de seguridad los paquetes contienen un texto desafío.

La seguridad a nivel de gateway y servidor contienen dos mecanismos. Uno es el de **Token** útil para acceder a los dispositivos y realizar cambios. El otro es la **Clave** que permite procesar los datagramas recibidos en cada dispositivo o etapa dentro de la cadena de avance la aplicación distribuida.

## Esquema



## Requisito Funcional

ID	Descripción	Referencia
RF05	Validar en la aplicación distribuida	<a href="#">ER</a>

### 3.6. RF06: Almacenamiento de datos

#### Descripción

El dispositivo acumula datos en 4 categorías.

#### 1- Datos provenientes de eventos normales

Son datos que se generan de los sensores, en general son de carácter volátil a excepción del backup de variables relevantes .

#### 2- Datos provenientes comunicación indirecta

Son datos que se originan en nodos adyacentes y corresponden a tablas de rutéo extendidas, datagramas en modo indirecto que serán enviadas según vayan despertando los destinatarios, son de carácter volátil.

#### 3- Datos de configuración

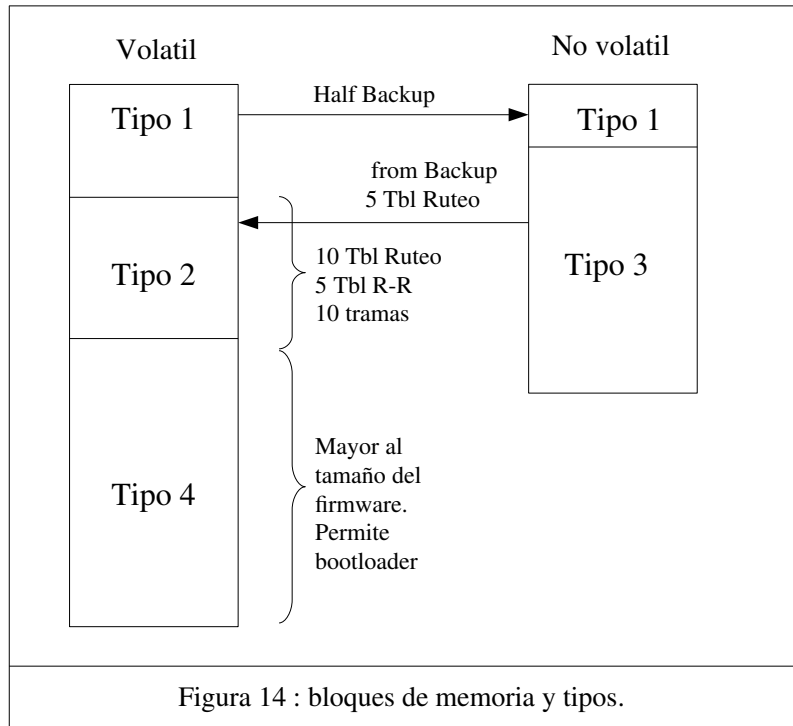
Son datos que se ajustan en el inicio del sistema para poder operar en la aplicación distribuida y la red, MAC, Id, Tk0, Tk1, key, slot y tablas de rutéo inicial, son de carácter no volátil.



#### 4- Datos provenientes del administrador

Son datos que tienen como objetivo modificar el comportamiento del sistema, bloques de datos para la aplicación de bootloader, datos a transmitir a la aplicación adjunta del dispositivo, secuencia de operaciones que ejecutan actuadores o drivers .

#### Esquema



#### Requisito Funcional

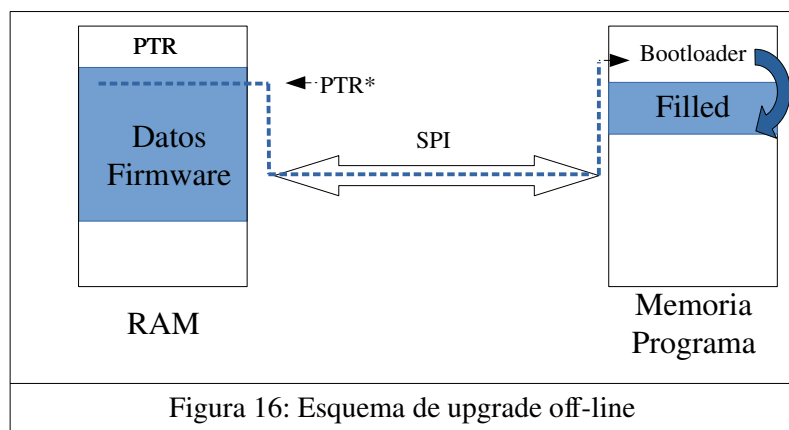
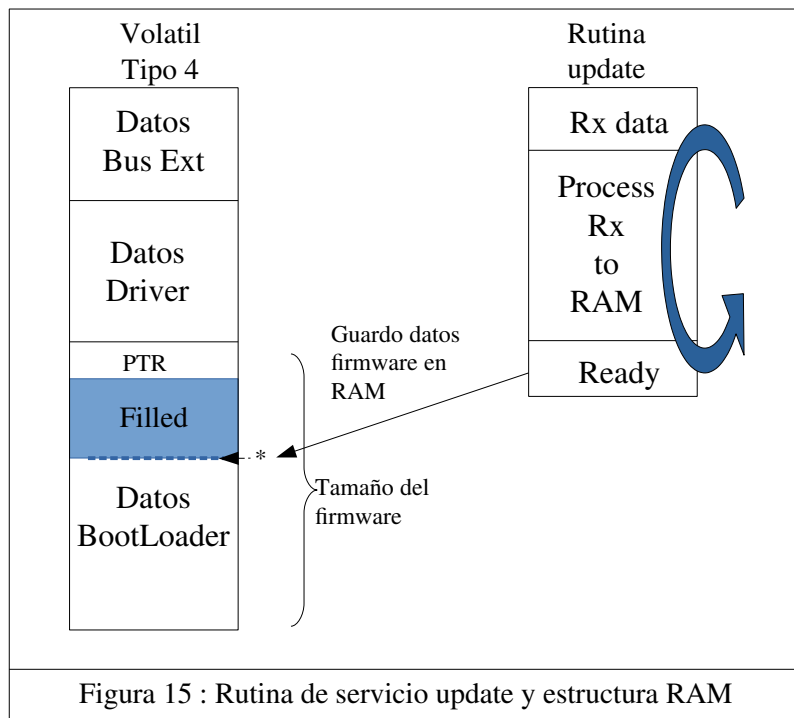
ID	Descripción	Referencia
RF06	Almacenamiento	<a href="#">ER</a>

### 3.7. RF07: Actualización remota

#### Descripción

El sistema requiere actualizarse para cumplir los requisitos funcionales iniciales que no se encuentran implementados en el prototipo y corresponden durante la etapa de PMV y el objetivo de mejora continua del producto y servicio.

La actualización remota tiene diferentes patrones de diseño, la opción que se implementara es modo off-line. Este requiere un gran bloque de memoria externa RAM, esto permite la actualización completa del MoTe.



### 3.8. RF08: Almacenar variables normales

El dispositivo requiere almacenar una serie de variables que son de funcional importancia. Estas variables son las que dan sentido a la razón del diseño del dispositivo, sin ellas el sistema no cumple su funciones principales.

Las variables se discriminan según su utilidad siendo eventos funcionales, eventos de management, de comunicaciones.

Funcionales	Management	Comunicaciones
<b>Medidas</b> <b>Fecha de medidas</b>	<b>Timestamp</b> <b>Clave</b> <b>Token</b> <b>Secuencia</b> <b>Slot</b> <b>Calendario</b>	<b>Ruido Rx</b> <b>Potencia Tx</b> <b>Retransmisiones</b> <b>BSS</b> <b>ID</b>

### Requisito Funcional

ID	Descripción	Referencia
RF08	Variables Normales	<a href="#">ER</a>

### 3.9. RF09: Enviar datos automáticamente a servidor

El dispositivo en su estado permanente se encuentra en modo **Normal**, el cual tiene definidos las variables de periodo de inactividad, periodos entre reportes, cantidad de variables normales designadas a transmitir.

Como los transceptores y las pilas que controlan estos dispositivos son para aplicaciones no intensivas, en caso de emplear grandes cantidades de dispositivos existe la posibilidad de colisiones en las comunicaciones. Los dispositivos empleados tienen mecanismos de CSMA/CA para el caso de MiWi y en LoRa-A unicamente Aloha.

Para no modificar estos protocolos se opto por incorporar una capa funcional entre estos protocolos y la aplicación para ambos.

En el caso de LoRa se implementaría una capa de red que a diferencia de MiWi no esta definida en su pila de protocolos.

Consiste en incorporar ranuras de transmisión, probabilidad de transmitir, back off en caso de colisiones o no ASK y diferimiento en caso de imposibilidad de comunicación en n intentos llevando las comunicaciones pendientes a un periodo común auxiliar o de emergencia empleado unicamente por las estaciones en estas circunstancias.

Si bien no es una mejora significativa para MiWi solo se incorpora la probabilidad de transmitir, para el caso de LoRa la mejora es significativamente mayor que Aloha puro.

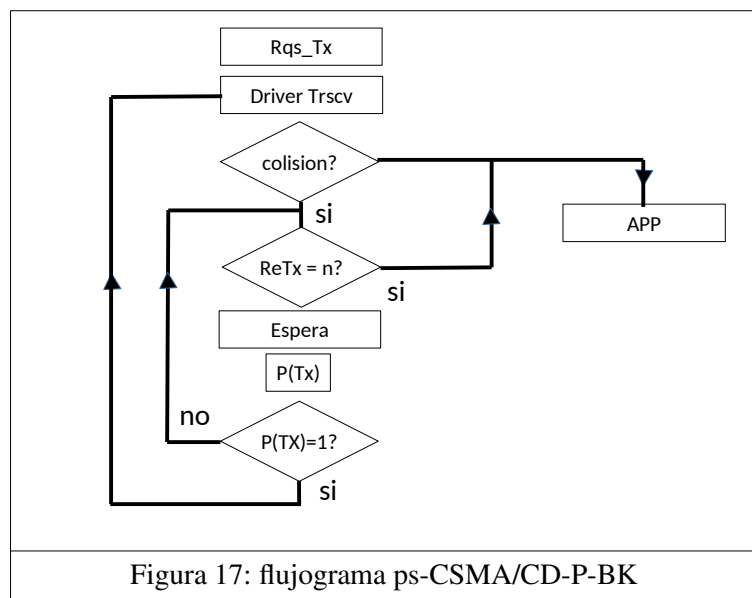
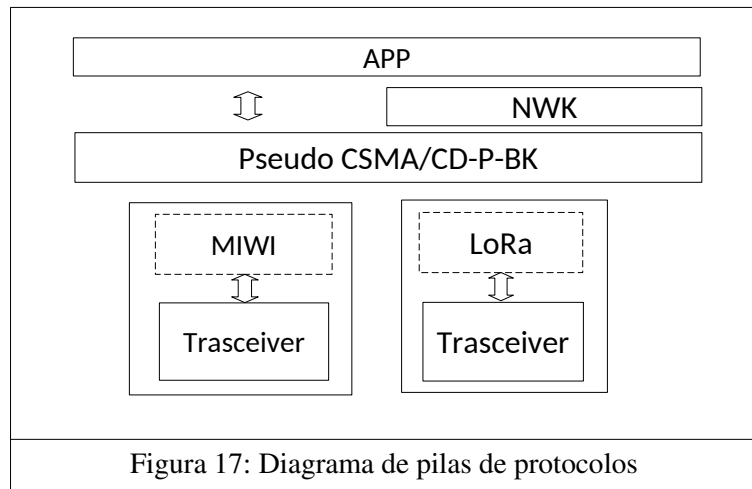
También se opto reducir la cantidad de reportes diarios a un máximo de 24 y un mínimo de 2.

La razón es que existen 1024 dispositivos por BSS en MiWi como se proyecta contar con 4 PA condición que reduciría los tiempos libres de colisiones. 4100 dispositivos deberían compartir el medio, menos de 1s entre transmisiones si se llevara a un esquema ordenado. Esto no tiene en cuenta el segundo método de funcionamiento del dispositivo adicional en modo **Continuo**.

Entonces a modo de seguridad se consideran periodos de Slots de 10s en las cuales los dispositivos realizan hasta n intentos de comunicacion bajo un esquema de back-off y probabilidad p de transmitir.

Para el caso de LoRa el esquema es aun mas exigente ya que no existe CSMA, la cantidad de dispositivos por PA es significativamente mayor, la longitud de la tramas en mayor y menores velocidades de transmisión.

Lo que se intenta alcanzar en modo **Normal** es un medio pseduo libre de contienda al definir y planificar cuando las estaciones pueden comunicarse a diferencia del estado **Continuo** que compite por el medio con las estaciones que se encuentran en modo **Normal**



### Requisito Funcional

ID	Descripción	Referencia
RF09	Reportes automáticos	<a href="#">ER</a>

### 3.10. RF10: Interacción y reacción con administrador

La interacción entre el dispositivo **MoTe** y el administrador durante el modo **Continuo** se efectúa mediante las datagramas tipo control. Estas datagramas tienen una rutina de procesamiento que determinan el comportamiento del dispositivo ajustando variables del sistema, colocando datos en buffers de busses externos e internos.

RF10. 01	Estado normal y Estado Continuo
RF10. 02	Ranura Transmisión
RF10. 03	Colocar byte en buffer de salida
RF10. 04	Leer byte en buffer de entrada
RF10. 05	Ajustar potencia del transmisor
RF10. 06	Cambiar canal de comunicación
RF10. 07	Inducir reinicio del sistema.
RF10. 08	Cambiar BSS
RF10. 09	Estado Continuo
RF10. 10	Bootloader

### Requisito Funcional

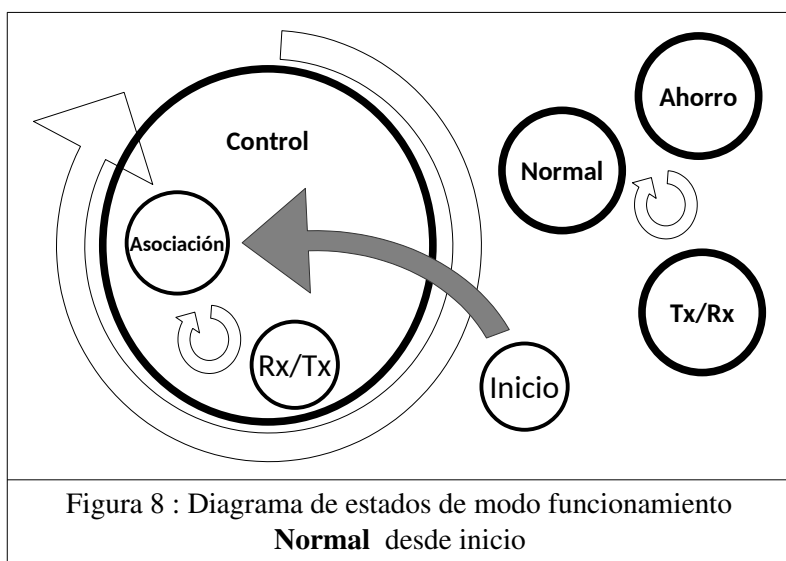
ID	Descripción	Referencia
RF10	Opciones de Control	<a href="#">ER</a>

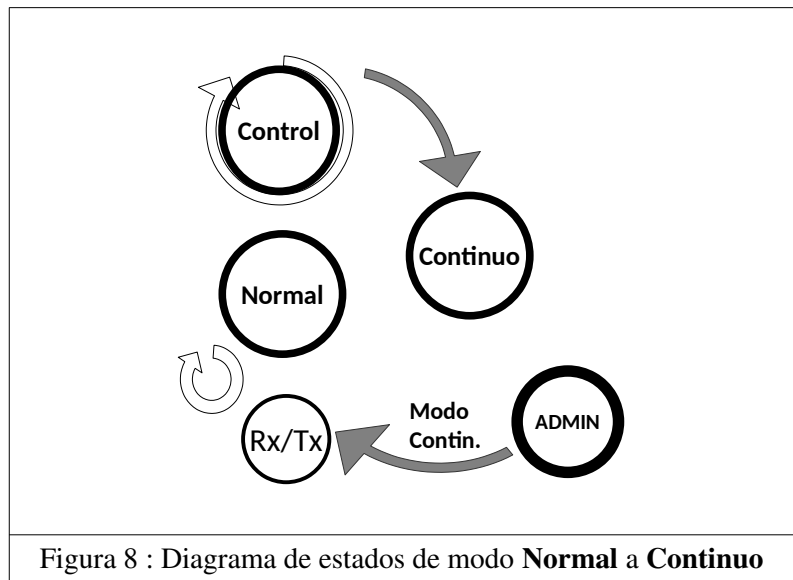
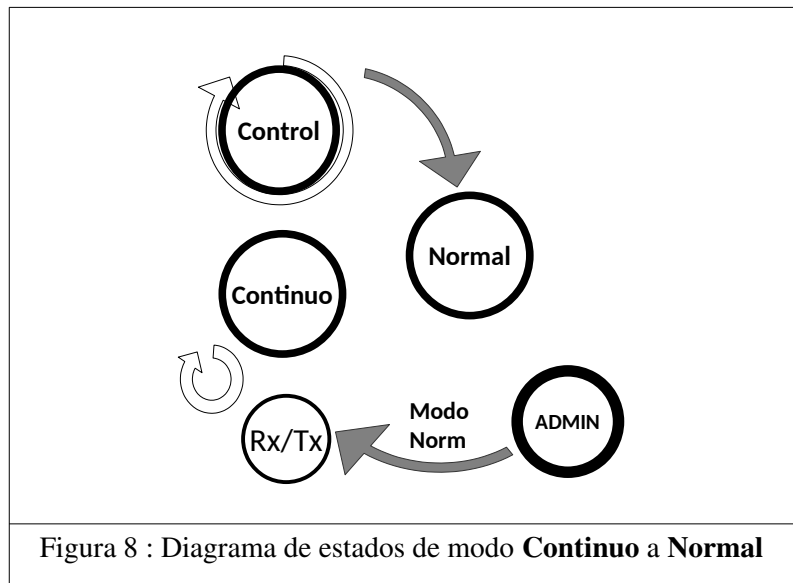
#### RF10.1: Requerimientos de control: Estado normal y Estado Continuo

Durante el inicio del sistema, el dispositivo se encuentra en estado de **asociación**. Como es una situación particular no se considera a esta como los casos **Normal** y **Continuo** si no que dentro del estado de **Control**.

Una vez asociado el MoTe el estado de **Control** ajusta el sistema a modo **Normal**.

Cuando el sistema se encuentra en estado **Continuo** y el administrador enviá la orden de pasar a estado **Normal** es interpretada por el estado **Control** y ejecutado por este.



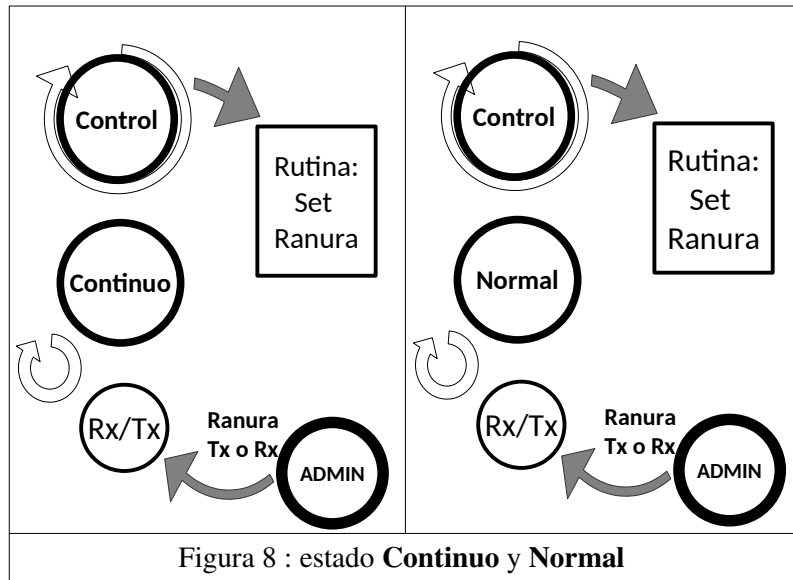


### RF10.2: Requerimientos de control: Ranura de Tx y Rx.

Un ajuste de control importante para evitar colisiones es asignar un esquema de reportes ordenado por el administrador de la red esto también es útil para que las estaciones se sincronicen y realicen comunicaciones indirectas.

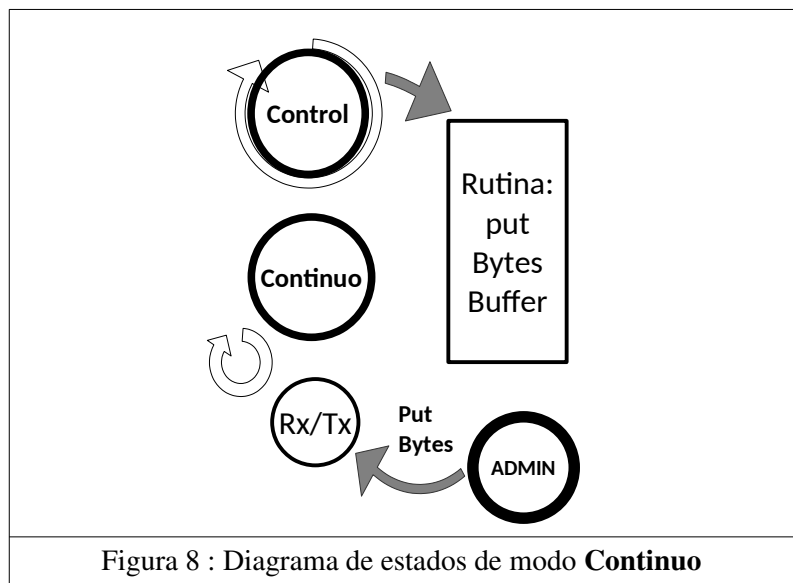
Las ranuras tienen por defecto una ventana de tiempo de 10 segundos. La ranura nº 0 corresponde a los primeros 10 s de las 00:00hs.

Existe un doble comando para modificar este mecanismo, se encuentra en las datagramas normales de respuesta y como una orden en las datagramas de control.



### RF10.3: Requerimientos de control: Colocar byte buffer salida de control

El administrador puede controlar el bus externo al colocar bytes en un buffer especial. De este modo puede efectuar acciones de control sobre la aplicación externa al MoTe.

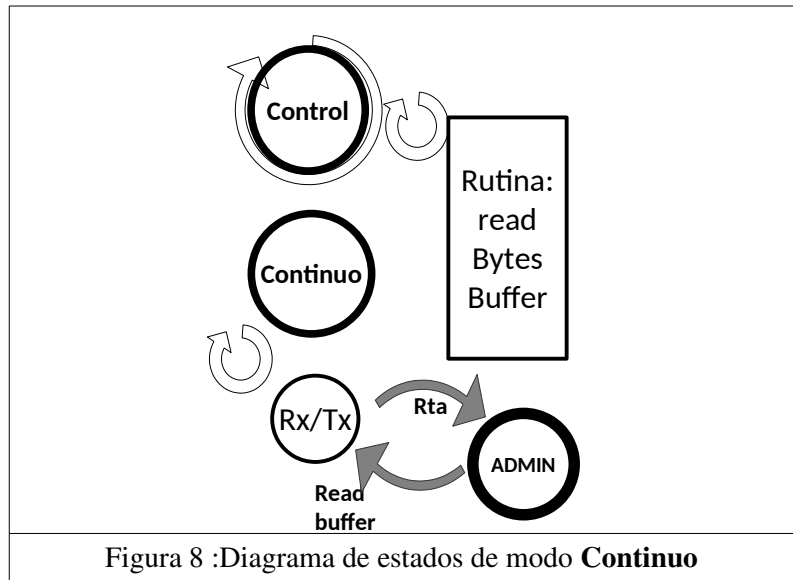


### RF10.4: Requerimientos de control: Leer byte buffer entrada de control

El administrador puede leer el bus externo mediante dos modos de operación:

- 1- Luego de haber recibido la una orden de escritura de buffer de salida y recuperar los datos de respuesta en un buffer de recepción.
- 2- Leer el buffer de recepción disponible.

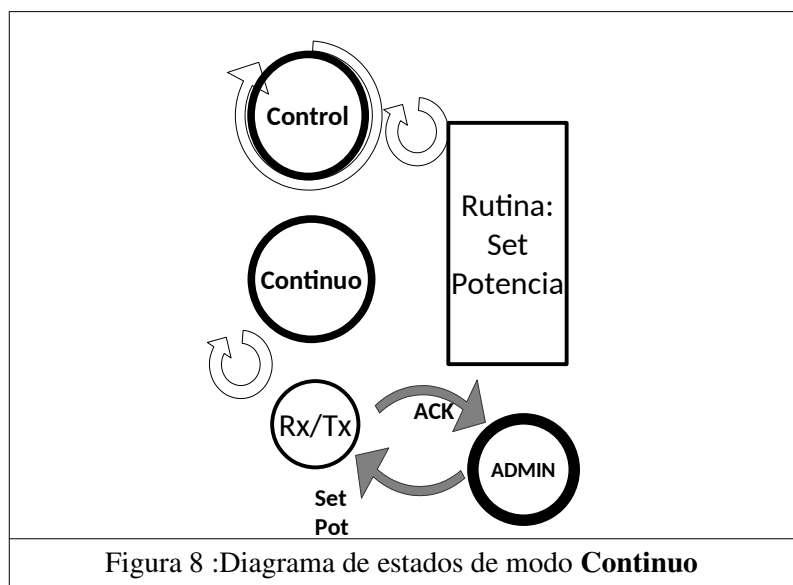
De este modo las acciones de control sobre la aplicación externa al MoTe se pueden monitorearse en tiempo real.



#### RF10.5: Requerimientos de control: Aumentar potencia Tx

El administrador puede aumentar la potencia de transmisión de un MoTe a fines de establecer una ruta para que un dispositivo lejano pueda tener acceso a la red.

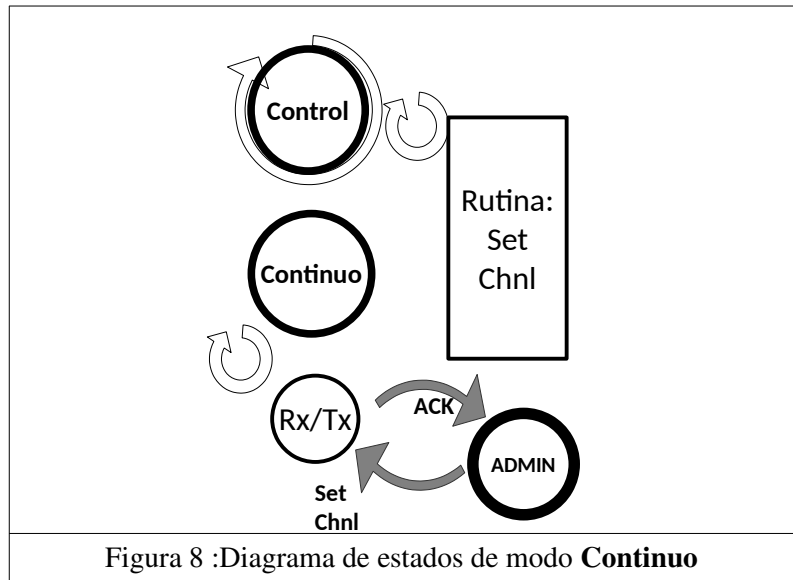
También es útil para el caso de reducir las retransmisiones, si bien el dispositivo cuenta con un mecanismo de auto ajuste a un mínimo frente al PA o a un MoTe próximo que lo asista.



#### RF10.6: Requerimientos de control: Cambiar canal TX

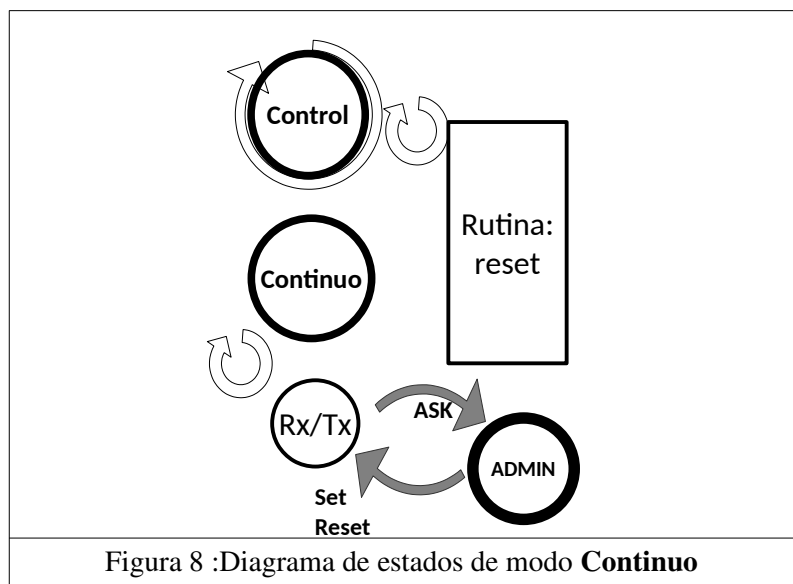
El administrador puede cambiar el canal de transmisión de un MoTe a fines de establecer una ruta con menor probabilidad de error o saturada por colisiones.





#### RF10.7: Requerimientos de control: Aplicar Reset

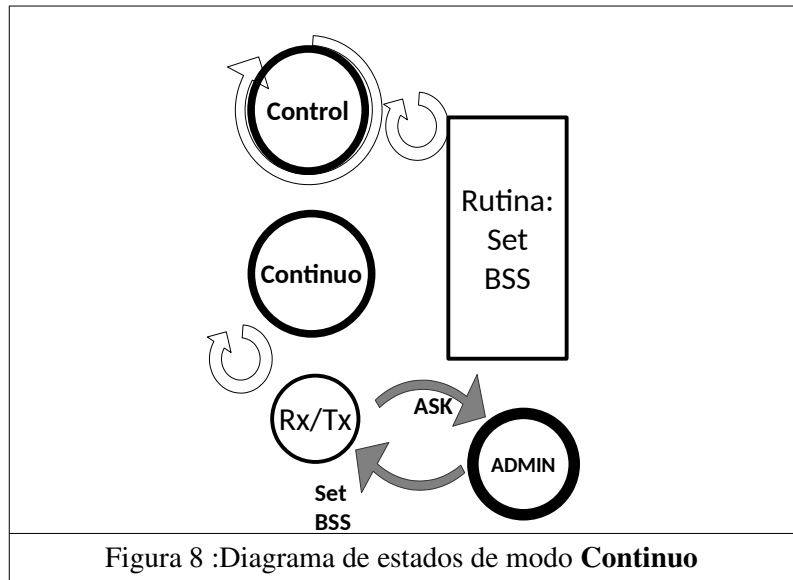
El administrador puede aplicar un reset al dispositivo con el fin de reiniciar el equipo de aplicación adjunta o simplemente con fines de control.



#### RF10.8: Requerimientos de control: Cambiar de red

El administrador puede cambiar el BSS de un MoTe a fines de establecer una ruta o balancear la carga sobre un PA.

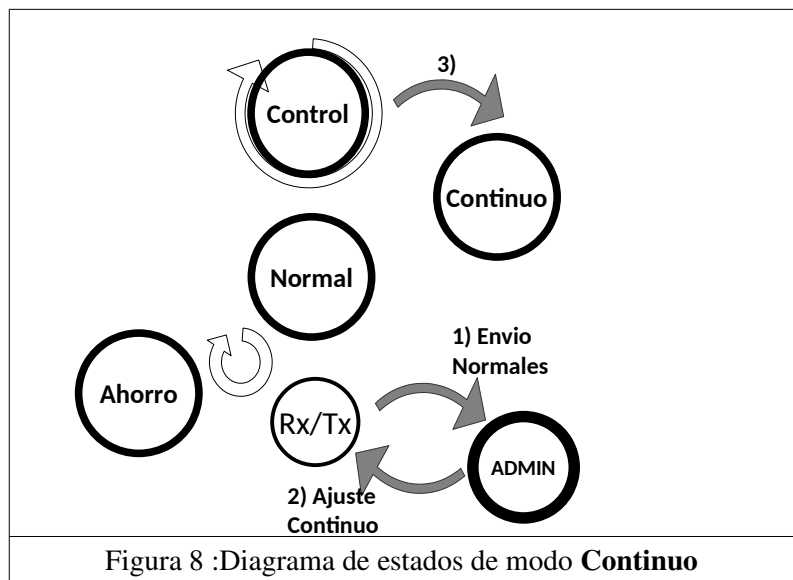
También es útil para el caso existir zonas de cobertura múltiple donde se desee organizar la red bajo un patrón determinado.

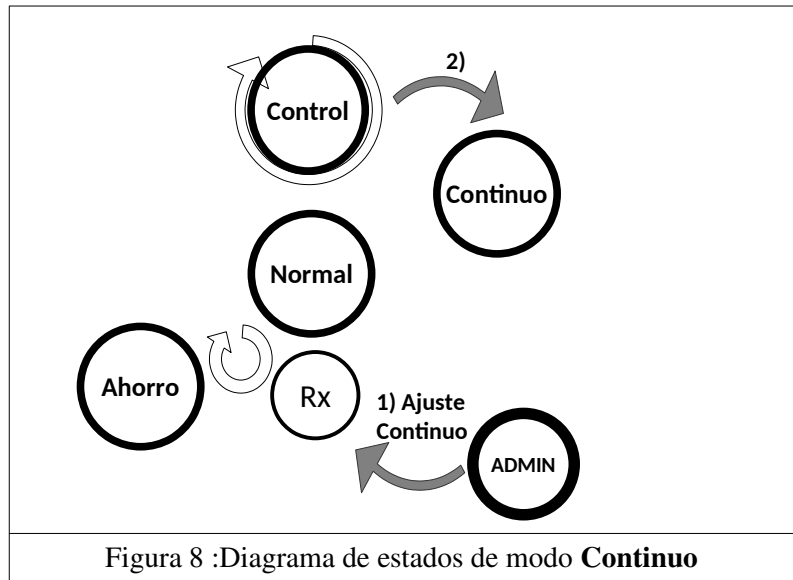


### RF10.9: Requerimientos de control: Estado modo continuo

El dispositivo entra en estado de funcionamiento continuo, sin ahorro de energía y respuestas a demanda del servidor. Se establece conexión bajo contienda por acceso al medio, la comunicación pseudo tiempo real.

El estado antecede al modo **Continuo** es el modo **Normal**. La estación recibe una solicitud de cambio de estado a **Continuo** cuando recibe un ACK por su transmisión en modo **Normal** o solo caso de conexiones directas cuando se encuentre en periodo de modo escucha activa luego de salir de ahorro de energía.





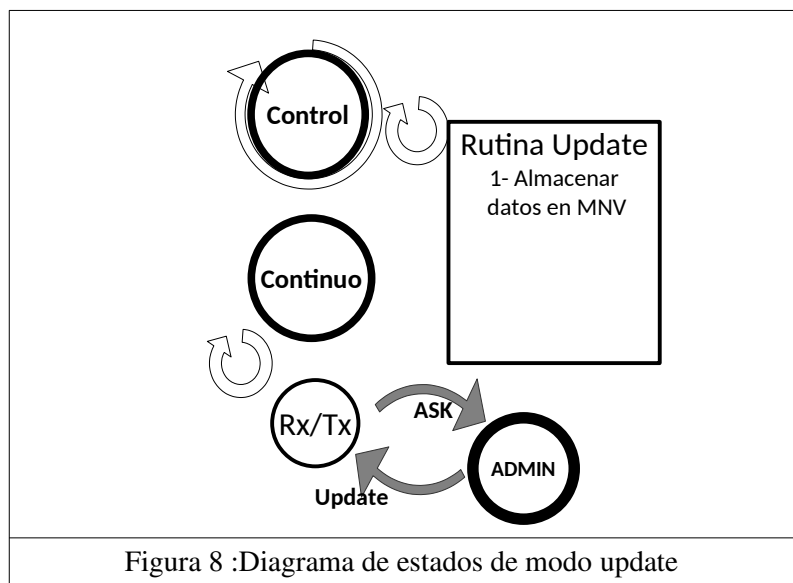
#### RF10.10: Requerimientos de control: Bootloader

El administrador puede aplicar un cambio de firmware mediante un upgrade al dispositivo.

Para cumplir con este requisito una porción de memoria de programa del MCU debe ser destinada para alojar un programa que solo tiene como fin realizar el upgrade, se denominada Bootloader.

La primer etapa del proceso se denomina update donde el administrador envía el nuevo firmware y se almacena en la memoria volátil externa. Una vez completada esa instancia el administrador determina cuando realizar el upgrade.

Cuando el administrador ordena iniciar el proceso de upgrade el dispositivo realiza un autoreset. Luego de un arranque del MCU con características particulares, el Bootloader comienza a ejecutarse.



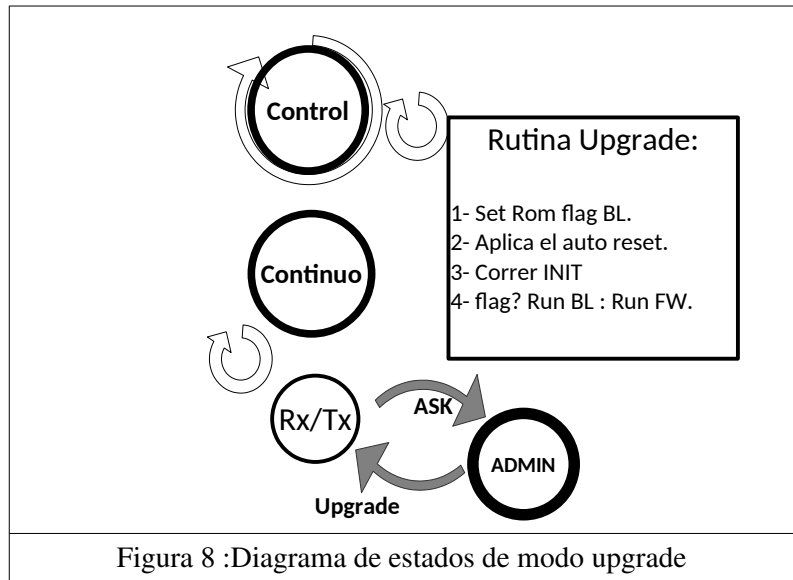


Figura 8 :Diagrama de estados de modo upgrade

#### 4. Requisitos no funcionales

##### Cualidades de ejecución.

##### RNF01: Requisitos de rendimiento

- Debe reportarse cada ranura de tiempo T asignado.
- Debe poder realizar un boot-reset en 3s.
- Debe contar con CSMA/CA-p[2] slot 3s.
- Debe contar con back-off 3s.
- Debe poder administrar una tabla de 3 adjuntos.
- Debe tener un consumo de corriente eléctrica promedio a operación normal menor a  $10^{-5}$  A.
- Debe tener capacidad **administrar** almacenamiento de 38kB de datos.

##### RNF02: Seguridad

- Los datos deben estar encriptados AES.
- El acceso a dispositivo requiere token y clave de 2 bytes.
- El acceso a la red requiere clave, 16 caracteres texto desafío.
- El dispositivo tiene identificación única.
- El sistema comunica al administrador por uso no autorizado.

##### RNF03: Fiabilidad

- El sistema debe almacenar datos durante 7 días, a razón de 400 bytes por día.
- El sistema debe recuperarse luego de una falla y comunicar al administrador.
- El sistema tiene una detección de fallas cada 4 hs.

##### RNF04: Disponibilidad

- El sistema se accede remotamente al menos hasta 100 m de distancias en línea de vista sin obstáculos.
- El sistema requiere mantener autonomía por 3 años para uso normal.

- El sistema requiere mantener autonomía por 3 meses para uso de tiempo completo con 2 baterías AAA 1,5V-1000mAh.

### **Cualidades evolución.**

#### **RNF05: Mantenibilidad**

- El sistema debe poder actualizarse remotamente en 10 s.
- Mantenimiento remoto.
- Se genera información de estado gral. del sistema en cada comunicación.
- Se genera información específica del sistema a demanda remota, variables no normales dependientes de la aplicación en campo, se acceden de lectura escritura como registros y bobinas en un PLC.

#### **RNF06: Extensibilidad**

- La red debe identificar 1028 dispositivos distintos.
- El dispositivo debe cambiar de red ante fuera de servicio de su PA luego de 4 hs.

#### **RNF07: Usabilidad**

- El dispositivo debe contar con manual de especificación técnica de 10 hojas.
- El dispositivo debe contar con detalles de fallas comunes y de reemplazo.
- El dispositivo tiene indicado su MAC.

#### **RNF08: Portabilidad**

- El código fuente debe ser escrito en C estándar 99.
- La API HAL debe estar implementada basada en objetos.
- La estructura del programa debe ser controlada por eventos.
- Se debe respetar la implementación estándar POSIX.

### **Otros requisitos**

#### **RNF09 Externos**

- Los datos se deben presentar de modo que faciliten la conversión a formato JSON hasta nivel 4.